

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

**In the Claims:**

No claims are added or canceled.

Claims 1—26 are pending and are listed as follows:

1           1. (Previously Presented) A method for processing a database query,  
2 comprising:

3           partially pre-aggregating records in a database according to a single  
4 grouping column to provide a result that contains at least two records having like  
5 grouping column values; and

6           aggregating records derived from the partial pre-aggregation to provide a  
7 result that contains records having unique grouping column values.

8  
9           2. (Original) The method as recited in claim 1, wherein the partially  
10 pre-aggregating further comprises:

11           maintaining a record store in memory, the record store having one record  
12 for each different grouping column value encountered in the operation;

13           receiving a new record;

14           combining the new record with a record having the same grouping column  
15 value, if such a record exists; and

16           adding the new record to the record store in the memory if there is no record  
17 in the record store that has the same grouping column value as the new record.

18  
19           3. (Original) The method as recited in claim 2, further comprising:

20           adding additional new records to the record store until the record store  
21 reaches a capacity such that it can accept no new records; and

22           outputting one or more records from the record store to a subsequent  
23 database operator.

1           4.     (Original) The method as recited in claim 3, wherein after the one or  
2 more records have been output to the subsequent database operator, the adding and  
3 outputting are repeated until there are no new records to process.

4  
5           5.     (Original) The method as recited in claim 4, wherein any records  
6 remaining in the record store after there are no new records to process are output to  
7 the subsequent database operator.

8  
9           6.     (Original) The method as recited in claim 3, wherein the subsequent  
10 database operator is a join.

11  
12           7.     (Original) The method as recited in claim 1, further comprising  
13 estimating the costs and benefits of the partial pre-aggregation, and partially pre-  
14 aggregating the records only if the estimating indicates that the benefits are greater  
15 than the costs.

16  
17           8.     (Original) The method as recited in claim 1, wherein the partially  
18 pre-aggregating includes utilizing a hashing function.

19  
20           9.     (Original) The method as recited in claim 1, wherein the partial pre-  
21 aggregating creates a record store in memory, and wherein the method further  
22 comprises utilizing the record store in memory for one or more other database  
23 operators.

1           10.   (Original) One or more computer-readable media having computer-  
2 executable instructions that, when executed by a computer, perform the method  
3 recited in claim 1.

4  
5           11.   (Original) A computer programmed to perform the method recited in  
6 claim 1.

7  
8           12.   (Previously Presented) A relational database system, comprising:  
9 memory for storing a record store, the memory having a portion available  
10 for query processing;

11           a query processor coupled to the memory to process a query on the record  
12 store according to a single grouping column, the query processor being configured  
13 to partially pre-aggregate the record store to provide a result that contains at least  
14 two data records that have like grouping column values; and

15           the query processor being further configured to aggregate data records  
16 resulting from the partial pre-aggregation to provide an aggregation result that  
17 contains data records, no two of the data records having the same grouping column  
18 value.

1           13.   (Previously Presented) The relational database system as recited in  
2 claim 12, wherein the query processor being configured to partially pre-aggregate  
3 the record store further comprises the query processor being configured to:

4           maintain a record store in the volatile memory, the record store having one  
5 record for each different grouping column value encountered in the partial pre-  
6 aggregation;

7           receive an input record from the non-volatile memory;

8           combine the input record with a record in the record store that has the same  
9 grouping column value, if there is such a record; and

10          adding the input record to the record store if there is no record in the record  
11 store that has the same grouping column value as the input record.

12  
13          14.   (Original) The relational database system as recited in claim 13,  
14 wherein the query processor is further configured to:

15          add additional input records to the record store in memory until the record  
16 store reaches a capacity such that it can accept no more input records; and

17          output the records in the record store to a subsequent database operator.

18  
19          15.   (Original) The relational database system as recited in claim 14,  
20 wherein the query processor is configured to:

21          continually add input records to the record store; and

22          output one or more records from the record store to a subsequent database  
23 operator when the record store reaches a capacity such that it can accept no more  
24 new records, or whenever there are no new records to process.

1           16.    (Original) The relational database system as recited in claim 12,  
2 wherein the query processor is further configured to perform a join on the records  
3 resulting from the partial pre-aggregation.  
4

5           17.    (Original) The relational database system as recited in claim 12,  
6 wherein the query processor is further configured to create a record store as a  
7 result of the partial pre-aggregation and utilize the record store in processing of  
8 another database operator.  
9

10          18.    (Original) The relational database system as recited in claim 12,  
11 further comprising:

12               a query optimizer configured to estimate the costs and benefits of the query  
13 processor performing a partial pre-aggregation; and

14               wherein the query processor performs the partial pre-aggregation only if the  
15 query optimizer indicates that the benefits of the partial pre-aggregation are greater  
16 than the costs of the partial pre-aggregation.  
17

18          19.    (Original) The relational database system as recited in claim 12,  
19 wherein the query processor is further configured to utilize hashing to perform the  
20 partial pre-aggregation.  
21

22          20.    (Original) The relational database system as recited in claim 12,  
23 wherein the query processor is further configured to utilize hashing and  
24 partitioning to perform the partial pre-aggregation.  
25

1           21. (Currently Amended) A relational database computer program  
2 embodied on a computer-readable medium, comprising:

3           partial pre-aggregation code to partially pre-aggregate data records  
4 according to grouping column values in a single grouping column to provide a  
5 partial pre-aggregation result having two or more records having like grouping  
6 column values; and

7           aggregation code to aggregate data records in the partial pre-aggregation  
8 result to provide an aggregation result having records with unique grouping  
9 column values.  
10

11           22. (Original) The relational database computer program as recited in  
12 claim 21, wherein the partial pre-aggregation code is designed to:

13           maintain a record store in memory, the record store having one record for  
14 each different grouping column value encountered in the partial pre-aggregation;

15           receive a new record;

16           combine the new record with a record in the record store having the same  
17 grouping column value, if such a record exists;

18           add the new record to the record store no record in the record store that has  
19 the same grouping column value as the new record;

20           continuously add additional new records to the record store until the record  
21 store has reached a record store capacity;

22           output one or more records from the record store to a subsequent database  
23 operator when the record store has reached the record store capacity; and

24           output the records in the record store when there are no new records to  
25 process.

1  
2 23. (Original) The relational database computer program as recited in  
3 claim 22, further comprising database operator code that utilizes the record store  
4 for input.

5  
6 24. (Currently Amended) A relational database computer program stored  
7 on a computer-readable medium, the relational database computer program  
8 comprising computer-executable instructions that, when executed on a computer,  
9 perform the following steps:

10 receiving a stream of input records;  
11 aggregating the input records in the stream according to a single grouping  
12 column as it is received to create a record store;  
13 joining records in the record store with other data;  
14 aggregating the records output from the join; and  
15 wherein the records output from the join include at least two records that  
16 have an identical grouping column value in the single grouping column.

17  
18 25. (Original) The relational database computer program as recited in  
19 claim 24, wherein:

20 the record store has a capacity that is less than the number of records in the  
21 stream of input records; and

22 the aggregating each input record is performed until the record store reaches  
23 capacity.



1        26. (Original) The relational database computer program as recited in  
2 claim 24, further comprising computer-executable instructions that, when executed  
3 by a computer, perform the following steps:

4        determining if it is optimal to aggregate the input records prior to  
5 performing the join; and

6        performing the aggregation prior to the join only if a determination is made  
7 that it is optimal to perform an aggregation prior to the join.